

Einleitung

Dieses Buch wendet sich an Entwickler, die Geräte entwerfen und programmieren, die die USB-Schnittstelle (Universal Serial Bus) verwenden. Das Ziel dieses Buches besteht darin, Ihnen USB vorzustellen und Sie dabei zu unterstützen, Ihre Geräte möglichst schnell und einfach betriebs- und kommunikationsbereit zu machen.

Die USB-Schnittstelle ist für eine große Bandbreite von Peripheriegeräten flexibel genug. Zu den Standardperipheriegeräten, die USB nutzen, zählen Maus, Tastatur, Festplatte, Drucker, Scanner und Audio/Videogeräte. USB eignet sich auch für Datenerfassungsgeräte, Steuerungssysteme und andere Geräte mit spezialisierten Funktionen bis hin zu Prototypen oder Unikaten.

Um ein Gerät mit einer USB-Schnittstelle entwickeln zu können, müssen Sie etwas über die Arbeitsweise der Schnittstelle wissen, darüber hinaus, welche Aufgaben die Geräte-Firmware übernehmen muss, um mit dem Bus kommunizieren zu können, und Sie müssen wissen, welche Klassentreiber und andere unterstützende Funktionen auf den Host-Computern zur Verfügung stehen, an die Ihr Gerät angeschlossen werden soll. Die richtige Auswahl der Geräte-Hardware, der Geräteklasse, der Entwicklungswerkzeuge und Verfahren kann Ihnen über weite Strecken dabei helfen, Pannen zu vermeiden und die anstehenden Aufgaben zu vereinfachen.

Wenn Sie sich mit der Entwicklung von USB-Geräten, dem Schreiben der Firmware von USB-Geräten oder dem Schreiben von Anwendungen zur Kommunikation mit USB-Geräten befassen, wird Sie dieses Buch dabei unterstützen.

Was Sie in diesem Buch finden

Hier sind einige Fragen, die dieses Buch beantwortet:

- *Wie kommunizieren USB-Geräte?* Die USB-Schnittstelle kann zunächst durchaus ein wenig entmutigend sein. Die USB-Spezifikation umfasst mehr als 600 Seiten, ohne dabei Klassenspezifikationen und andere zusätzliche Dokumente mitzuzählen. Dieses Buch versucht nicht, die Aussagen der Spezifikation zu wiederholen. Stattdessen konzentriert es sich auf das Wissen, das Sie benötigen, damit Ihre Geräte effizient und zuverlässig kommunizieren können.

- *Wie kann ich entscheiden, ob mein Gerät eine USB-Schnittstelle verwenden sollte?* USB ist nicht für alle Anwendungen die erste Wahl. Finden Sie heraus, ob für Ihre Entwicklung USB oder eine andere Schnittstelle verwendet werden sollte. Es ist jedoch recht wahrscheinlich, dass Sie sich für USB entscheiden. Und wenn das der Fall ist, dann werden Sie hier erfahren, wie Sie sich für eine der drei USB-Geschwindigkeiten und einen der vier Transfertypen für Ihre Anwendung entscheiden können.
- *Welchen USB-Controllerchip sollte mein Gerät verwenden?* Alle USB-Geräte benötigen zur Implementierung der USB-Schnittstelle einen intelligenten Controller. Dutzende Hersteller bieten Controllerchips mit unterschiedlichen Architekturen und Eigenschaften an. Dieses Buch enthält Beschreibungen verbreiteter Chips und gibt Ihnen Tipps, die Sie bei der Auswahl eines Controllers gemäß der Anforderungen Ihres Projekts sowie Ihres eigenen Hintergrunds und Ihrer Präferenzen unterstützen sollen.
- *Wie kommunizieren Anwendungen mit USB-Geräten?* PC-Anwendungen greifen auf ein USB-Gerät zu, indem sie mit dem Gerätetreiber kommunizieren, der dem Gerät zugeordnet wurde. Einige Geräte können die zusammen mit Windows ausgelieferten Klassentreiber benutzen. Andere benötigen speziell angepasste Treiber. Dieses Buch wird Ihnen die verschiedenen Geräteklassen vorstellen und Ihnen dabei helfen zu entscheiden, ob sich eine der definierten Klassen für Ihr Gerät eignet. Wenn für Ihr Gerät ein speziell angepasster Treiber benötigt wird, dann werden Sie erfahren, was Sie zum Schreiben eines Treibers wissen müssen, welche Werkzeuge den Entwicklungsprozess beschleunigen können und welche Möglichkeiten bestehen, Treiber über andere Quellen zu beziehen. Beispielcode zeigt Ihnen, wie Sie Geräte in mit Visual Basic .NET und Visual C++ .NET geschriebenen Anwendungen erkennen und mit ihnen kommunizieren können.
- *Welche Firmware benötigt mein Gerät zur Unterstützung der USB-Kommunikation?* Erfahren Sie, wie Sie Geräte-Firmware schreiben können, mit deren Hilfe Ihr Gerät empfangene Anforderungen beantworten und Daten über den Bus austauschen kann.
- *Wie kann ich entscheiden, ob mein Gerät Bus-betrieben werden kann, oder ob es ein eigenes Netzteil benötigt?* Viele USB-Geräte können vollständig über den Bus mit Strom versorgt werden. Dann spricht man von »Bus-betriebenen« Geräten. Stellen Sie fest, ob dies bei Ihrem Gerät der Fall ist. Erfahren Sie, wie Sie dafür sorgen können, dass Ihr Gerät die USB-Auflagen zur Beschränkung der Stromaufnahme über den Bus erfüllt, wenn der Host-Computer den Bus in den Suspend-Status versetzt.
- *Kann ich andere USB-Peripheriegeräte an mein Gerät anschließen?* Finden Sie heraus, wie sich USB On-The-Go einsetzen lässt, damit Ihr Gerät als Host mit ein-

geschränkten Fähigkeiten eingesetzt werden kann, der auf andere USB-Peripheriegeräte zugreifen kann.

- *Wie kann ich dafür sorgen, dass mein Gerät problemlos kommuniziert?* Seitens des Geräts gehört zum Schreiben fehlerfreier Firmware das Wissen, dass Ihr Gerät die Anforderungen der USB-Spezifikation erfüllen muss. Auf Seiten des Host-Computers muss Windows über die zur Identifikation des Geräts und zum Auffinden eines Treibers erforderlichen Informationen verfügen, um mit dem Gerät kommunizieren zu können. In diesem Buch finden Sie Tipps, Beispielcode und Informationen zum Debuggen von Software und Hardware, die Sie bei diesen Aufgaben unterstützen sollen.

Um die Materialien in diesem Buch verstehen zu können, sind grundlegende Kenntnisse in einigen Bereichen hilfreich. Ich gehe davon aus, dass Sie gewisse Kenntnisse im Bereich digitaler Schaltungen, der Anwendungsprogrammierung für PCs und dem Schreiben eingebetteten Codes für Peripheriegeräte besitzen. Über USB müssen Sie hingegen gar nichts wissen.

Was ist neu in der dritten Auflage?

Seit der Veröffentlichung von *USB 2.0* hat sich rund um USB eine Menge getan. Zu den Ergänzungen zur USB-Spezifikation zählen viele aktualisierte und erweiterte Spezifikationen für Geräteklassen und der Zusatz *USB On-The-Go*. Viele neue Controllerchips wurden erhältlich. Neue Werkzeuge für die Fehlersuche und Kompatibilitätstests sind verfügbar. Die Unterstützung der USB-Geräteklassen unter Windows wurde verbessert. Und das .NET-Framework von Microsoft wurde zu einer beliebten Plattform zur Entwicklung von Host-Anwendungen.

Diese Entwicklungen führten zu dieser mittlerweile dritten vollständig überarbeiteten Version des USB-Buches. Das Material wurde komplett durchgesehen und von Anfang bis Ende aktualisiert, um diese und andere Entwicklungen im Zusammenhang mit der USB-Hardware und der USB-Programmierung zu erfassen.

Weitere Informationen, Aktualisierungen und Korrekturen

Wenn Sie mehr über die Entwicklung von USB-Geräten und der Software zur Kommunikation mit ihnen wissen wollen, dann können Sie meine englischen USB-Seiten auf der Website von Lakeview Research unter www.lvr.com besuchen. Dort finden Sie Codebeispiele, Links zu Artikeln, Produkten, Hilfsprogrammen und weitere Informationen, die im Zusammenhang mit der Entwicklung von USB-Geräten stehen. Wenn Sie einen Vorschlag, Code oder andere Informationen besitzen, die ich dort Ihrer Meinung nach veröffentlichen oder verlinken soll, dann können Sie mir dies auf englisch über jan@lvr.com mitteilen.

Korrekturen und Aktualisierungen zur englischen Auflage werde ich ebenfalls unter www.lvr.com veröffentlichen.

Danksagung

USB ist ein viel zu umfangreiches Thema, um ohne Hilfe darüber schreiben zu können. Daher muss ich vielen Personen danken!

Großen Dank schulde ich meinen Fachrezensenten, durch deren Rückmeldungen dieses Buch sehr gewonnen hat. (Dessen ungeachtet geht jeder Fehler im Buch ausschließlich auf mein Konto.)

Ich danke zunächst Paul E. Berg, stellvertretender Direktor von MCCI und Vorsitzender der Arbeitsgruppe Geräte beim USB-IF. Dank gebührt auch David Goll von der USB-IF-Arbeitsgruppe Videogeräte, Lucio DiJasio and Rawin Rojvanit von Microchip Technology, John Hyde von usb-by-example.com, Geert Knapen von der USB-IF-Arbeitsgruppe Audiogeräte, Walter Oney von Walter Oney Software und Marc Reinig von System Solutions.

Des Weiteren möchte ich folgenden Personen für ihre Unterstützung danken: Glenn M. Roberts von Cypress Semiconductor, Fred Dart und Keith Dingwall von FTDI Chip, Wendy Dee von Keil Software, Michael DeVault von DeVaSys Embedded Systems, Alan Lowne von Saelig Company, Laurent Guinnard von Ellisys, Rich Moran von RPM Systems Corporation und Bob Nathan von NCR Corporation.

Für die Hilfe und Unterstützung beim Verfassen der bisherigen Auflagen, auf die diese Auflage aufbaut, möchte ich Joshua Buerge, Gary Crowell, Dave Dowler, Mike Fahrion, John M. Goodman, Lane Hauck, David James, Christer Johansson, Kosta Koeman, Jon Lueker, Brad Markisohn, Amar Rajan, Robert Severson, Craig R. Smith und Dave Wright danken.

Ich hoffe, dass Ihnen das Buch von Nutzen sein wird. Anmerkungen sind jederzeit willkommen!

Jan Axelson

jan@lvr.com